PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-034848
(43)Date of publication of application : 07.02.1997

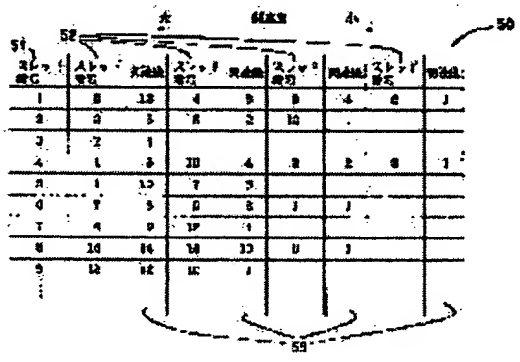
(51)Int.Cl. G06F 15/16
G06F 9/46
G06F 9/46

(21)Application number : 07-185733
(22)Date of filing : 21.07.1995

(71)Applicant : CANON INC
(72)Inventor : SUDO YOSHIAKI

(54) THREAD CONTROL METHOD AND INFORMATION PROCESSING SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a thread relation degree measurement method, a thread control method and an information processing system by conducting relation degree measurement so as to enhance a relation degree of a thread accessed to the same page in a short time thereby reducing recording area to record the relation degree.
SOLUTION: In the information processing system where plural information processing units are connected by a network and a thread is distributed on each information processing unit for the execution in a distribution task sharing in common a main storage in existence on distributed information processing units in shared by a distributed virtual share storage system, each thread is provided with storage areas 52, 53 storing the relation degree with other prescribed number of threads, and when a page access is given from a thread in the distributed task to a distributed virtual common share storage, a storage area 50 is updated, based on thread information possessing at present a page included in page management information and when a thread switching request comes, a thread operated on the same information processing unit is selected, based on the relation degree.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

特開平9－34848

(43)公開日　平成９年(1997)２月７日

(51)Int.Cl.⁶	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ06Ｆ 15/16	３７０		Ｇ06Ｆ 15/16	３７０Ｎ	
9/46	３４０		9/46	３４０Ｂ	
	３６０			３６０Ｂ	

審査請求　未請求　請求項の数12　ＯＬ　（全 11 頁）

(21)出願番号　特願平7－185733

(22)出願日　平成７年(1995)７月21日

(71)出願人　000001007
キヤノン株式会社
東京都大田区下丸子３丁目30番２号

(72)発明者　数藤　義明
東京都大田区下丸子３丁目30番２号　キヤノン株式会社内

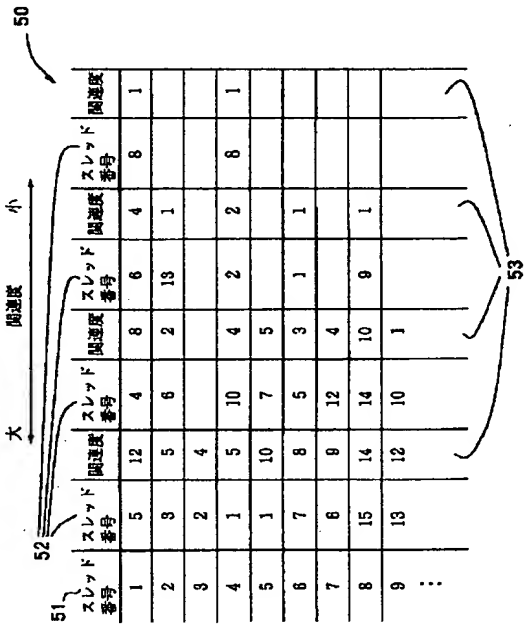
(74)代理人　弁理士　大塚　康徳　（外１名）

(54)【発明の名称】　スレッド制御方法及び情報処理システム

(57)【要約】

【課題】　短時間に同じページにアクセスするスレッドの関連度を高くするような関連度計測が行え、その関連度を記録する記録領域が小さくできるスレッド関連度計測方法、スレッド制御方法及び情報処理システムを提供する。

【解決手段】　複数の情報処理装置をネットワークで接続し、分散した情報処理装置上に存在するタスクの主記憶を分散仮想共有記憶方式によって共有する分散タスク内で、各情報処理装置上にスレッドを分散して実行させる情報処理システムにおいて、各スレッド５１に対して他の所定数のスレッドとの関連度を記憶する記憶領域（５２，５３）を用意し、分散タスク内のスレッドから分散仮想共有記憶へのページアクセスがあった場合に、ページ管理情報に含まれる該ページを現在所有しているスレッドの情報に基づいて、前記記憶領域５０を更新し、スレッドの切換え要求があった場合に、前記関連度に基づいて、同じ情報処理装置上で動作させるスレッドを選択する。



1

【特許請求の範囲】

【請求項１】 複数の情報処理装置をネットワークで接続し、分散した情報処理装置上に存在するタスクの主記憶を分散仮想共有記憶方式によって共有する分散タスク内で、各情報処理装置上にスレッドを分散して実行させる情報処理システムにおいてスレッドを制御するスレッド制御方法であって、

各スレッドに対して他の所定数のスレッドとの関連度を記憶する記憶領域を用意し、

分散タスク内のスレッドから分散仮想共有記憶へのページアクセスがあった場合に、ページ管理情報に含まれる該ページを現在所有しているスレッドの情報に基づいて、前記記憶領域を更新し、

スレッドの切換え要求があった場合に、前記関連度に基づいて、同じ情報処理装置上で動作させるスレッドを選択することを特徴とするスレッド制御方法。

【請求項２】 前記記録領域には、各スレッドに対して関連度の高い所定数のスレッドを選択して記録することを特徴とする請求項１記載のスレッド制御方法。

【請求項３】 前記ページ管理情報に該ページをアクセスした複数のスレッドの履歴情報を更新可能に記憶し、前記記憶領域の更新は該複数のスレッドの履歴情報に基づいて行われることを特徴とする請求項１記載のスレッド制御方法。

【請求項４】 前記記憶領域の更新はスレッドの更新と関連度の更新とを含むことを特徴とする請求項１乃至３のいずれか１つに記載のスレッド制御方法。

【請求項５】 複数の情報処理装置をネットワークで接続し、分散した情報処理装置上に存在するタスクの主記憶を分散仮想共有記憶方式によって共有する分散タスク内で、各情報処理装置上にスレッドを分散して実行させる情報処理システムにおいて、

各スレッドに対して他の所定数のスレッドとの関連度を記憶する記憶手段と、

分散タスク内のスレッドから分散仮想共有記憶へのページアクセスがあった場合に、ページ管理情報に含まれる該ページを現在所有しているスレッドの情報に基づいて、前記記憶手段の内容を更新する更新手段と、

スレッドの切換え要求があった場合に、前記関連度に基づいて、同じ情報処理装置上で動作させるスレッドを選択するスレッド選択手段とを備えることを特徴とする情報処理システム。

【請求項６】 前記記録手段には、各スレッドに対して関連度の高い所定数のスレッドを選択して記録することを特徴とする請求項５記載の情報処理システム。

【請求項７】 前記ページ管理情報に該ページをアクセスした複数のスレッドの履歴情報を更新可能に記憶し、前記更新手段は該複数のスレッドの履歴情報に基づいて前記記憶手段の内容を更新することを特徴とする請求項５記載の情報処理システム。

2

【請求項８】 前記更新手段はスレッドの更新と関連度の更新とを行うことを特徴とする請求項５乃至７のいずれか１つに記載の情報処理システム。

【請求項９】 複数の情報処理装置をネットワークで接続し、分散した情報処理装置上に存在するタスクの主記憶を分散仮想共有記憶方式によって共有する分散タスク内で、各情報処理装置上にスレッドを分散して実行させる情報処理システムにおけるスレッド間の関連度を計測するスレッド関連度計測方法であって、

各スレッドに対して他の所定数のスレッドとの関連度を記憶する記憶領域を用意し、

分散タスク内のスレッドから分散仮想共有記憶へのページアクセスがあった場合に、ページ管理情報に含まれる該ページを現在所有しているスレッドの情報に基づいて、前記記憶領域を更新し、

スレッドの切換え要求があった場合に、前記関連度に基づいて、スレッド間の関連度を算出して比較することを特徴とするスレッド関連度計測方法。

【請求項１０】 前記記録領域には、各スレッドに対して関連度の高い所定数のスレッドを選択して記録することを特徴とする請求項９記載のスレッド関連度計測方法。

【請求項１１】 前記ページ管理情報に該ページをアクセスした複数のスレッドの履歴情報を更新可能に記憶し、前記記憶領域の更新は該複数のスレッドの履歴情報に基づいて行われることを特徴とする請求項９記載のスレッド関連度計測方法。

【請求項１２】 前記記憶領域の更新はスレッドの更新と関連度の更新とを含むことを特徴とする請求項９乃至１１のいずれか１つに記載のスレッド関連度計測方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は分散した複数の情報処理装置の分散仮想共有記憶上で動作する分散タスク／スレッドモデルのプログラムにおいてスレッドを制御するスレッド制御方法及びその方法を実行する情報処理システムに関し、特にスレッド制御において各スレッド間の関連度を計測する関連度計測方法に関するものである。尚、ここで関連度計測方法とは、スレッド間の関連度の制御，算出及び比較を含むものである

【０００２】

【従来の技術】複数のプロセッサを１台の情報処理装置内にもつマルチプロセッサ情報処理装置上で、これら複数のプロセッサを単一のプログラムによって有効に活用可能な、タスク／スレッドモデルと呼ばれるプログラム形態が提案されている。これは、１つのプログラムをスレッドと呼ばれる複数の実行モジュールと、タスクと呼ばれる資源割り当ての単位に分割したモデルである。スレッドはプロセッサ資源の割り当て単位となり、その他の記憶空間資源などはタスクに割り当てられてタスク内

の全てのスレッドに開放されている。このようなタスク／スレッドモデルは、マルチプロセッサの情報処理装置において、プロセッサ資源を効率よく使用するプログラムのためのモデルである。

【０００３】さらに、通常のタスク／スレッドモデルのプログラム上で、利用者空間内でのコンテキスト切り換えやスレッド生成等を行なうことが可能な、利用者レベルスレッドが提案されている。これは、通常のタスク／スレッドモデルのスレッドの生成やコンテキスト切り換え等が、オペレーティングシステム・カーネル（ＯＳカーネル）へのシステムコールを必要とするので、速度が遅いという欠点を改善するために考えられたものである。すなわち、複数のコンテキストを持つことが可能で、かつ利用者空間内でのスレッド生成やコンテキスト切り換え等が高速に行なえることが利点である。このような利用者レベルスレッドに対して、従来のＯＳカーネルに制御されるスレッドをカーネルスレッドと呼ぶ。

【０００４】また、特殊な装置を持たずに、従来の主記憶管理装置と情報処理装置間のネットワークを制御してデータを転送することによって、複数の情報処理装置上のタスク間で仮想共有記憶を実現する分散仮想共有記憶方式を用いて、そのタスク内の全仮想記憶空間を共有し、その中で複数のスレッドを動作させる分散タスク／スレッドモデルが提案されている（B.Nitzberg and V.Lo, "Distributed SharedMemory: A Survey of Issues and Algorithms, " Computer,Vol.24,No.8,August 1991, pp.52-60.）。これは、複数の情報処理装置上の各タスク上の全主記憶を分散仮想共有記憶とし、各タスクが１個以上のスレッドを持った１つの分散したタスクと考えたものである。上述のタスク／スレッドモデルにおけるマルチプロセッサの情報処理装置を、複数の情報処理装置とネットワークによる接続に置き代え、これらの分散した資源を効率よく使用するためのものが、分散タスク／スレッドモデルである。また、分散仮想共有記憶はネットワークを主に固定長のページデータ転送で使用し、高速なネットワークを効率よく使用可能となる。

【０００５】

【発明が解決しようとする課題】本発明者は、このような分散仮想共有記憶で共有された主記憶上でタスク／スレッドモデルのタスクを動作させ、そのタスク内で利用者レベルスレッドを動作させるタスク／スレッド制御方式を提案した（特願平６－１６６８２４）。これは、分散仮想共有記憶によって分散した情報処理装置のどこからでもタスクの主記憶の空間がアクセスできることから、そのような主記憶上の利用者空間にコンテキストの保存を行なう利用者レベルスレッドは、分散したどの情報処理装置にも移動して動作可能となることを特徴としている。

【０００６】上述の分散仮想共有記憶では、各情報処理装置上の主記憶はその情報処理装置以外からはアクセスできないので、主記憶を一種のキャッシュとみなして、マルチプロセッサシステムのキャッシュの一貫性保持動作と同様な操作をして、各情報処理装置上の主記憶間の一貫性を保持する。この操作を行なうことによって、各情報処理装置が分散仮想共有記憶にアクセスした時の複数の情報処理装置の主記憶上のデータの一貫性が保証される。このような各情報処理装置の主記憶の一貫性保持のための操作として、一般的にWrite-Invalidate方式を用いる。このWrite-Invalidate方式で代表的なものとしてＭＥＳＩ(Modified,Exclusive,Shared,Invalid) 型方式がある。

【０００７】このようなWrite-Invalidate方式によって一貫性を保持する分散仮想共有記憶では、同じ分散仮想共有記憶のページを複数の情報処理装置内のプロセッサが同時にアクセスすると、ページスラッシングと呼ばれる頻繁なページの移動が発生し、ネットワークの輻輳を起したり情報処理装置の負荷が大きくなり、処理能力が極端に低下するという問題がある。

【０００８】そこで、更に本発明者は、分散タスク内のスレッド制御によって、同じ分散仮想共有記憶のページに同時期にアクセスするようなスレッドを同一のタスクで実行することによって、ページスラッシングを低減する方式を提案した（特願平６－１６６８２２）。これは、スレッドの分散仮想共有記憶へのアクセス情報によってスレッド間の関連度を計測し、関連度の高いスレッドは同一の情報処理装置上で動作させるような制御をすることでなされる。静的に各スレッド間の関連度が決定できるものでは、スレッド生成時に関連度の高いスレッドを同じ情報処理装置で動作させることでページスラッシングを低減する。動的に制御するものでは、スレッドの動作時のアクセス情報によってスレッド間の関連度を計測し、それに従ってスレッドの情報処理装置間での移送を行い、関連度の高いスレッドを同じ情報処理装置上で動作させることでページスラッシングを低減する。また、利用者レベルスレッドを利用する場合には、カーネルスレッドの移送をせずに、利用者レベルスレッドのスケジューリングによって関連度の高い利用者レベルスレッドを同じ情報処理装置上のカーネルスレッド上で動作させることが可能となり、ページスラッシングを低減することができる。

【０００９】上述の動的なスレッド間の関連度の計測では、分散仮想共有記憶の管理装置に対して、各スレッドが分散仮想共有記憶にアクセスした時にカーネルから通知されるページフォルトを用いて行なう。すなわち、分散仮想共有記憶の各ページを所有しているスレッドの情報を用いて、そのページでフォルトを起したスレッドとそのページを所有しているスレッドとの関連度を増加させるという操作を行なう。これによって、同じページにアクセスする頻度の高いスレッド間の関連度が上がり、ページスラッシングを低減するようなスレッド制御を行

なうことが可能となる。

【００１０】しかしながら、このような分散タスク内のスレッド間の関連度の計測を行なうためには、分散仮想共有記憶の管理装置（管理サーバ）かまたはスレッド管理ライブラリなどの記憶空間中に、各スレッド毎に他のスレッドとの関連度を保存しておく領域が必要となる。各スレッド毎に他の全てのスレッドに対する関連度を記憶しておく領域を用意すると、例えばスレッド数がＮであるとするとＮ×（Ｎ－１）／２の領域が必要となり、スレッド数が多くなるに従ってその領域が膨大になり、システム全体の記憶領域（この場合、分散仮想共有記憶管理サーバやスレッド管理ライブラリ内の記憶領域）を圧迫するという問題があった。

【００１１】本発明は、短時間に同じページにアクセスするスレッドの関連度を高くするような関連度計測が行え、その関連度を記録する記録領域が小さくできるスレッド関連度計測方法を提供する。又、本発明は、上記スレッド関連度計測方法を適用してスレッドを制御するスレッド制御方法及び情報処理システムを提供する。

【００１２】

【課題を解決するための手段】本発明によると、複数の情報処理装置をネットワークで接続した分散情報処理装置で、分散した情報処理装置上に存在するタスクの主記憶を分散仮想共有記憶によって共有する分散タスク内で各情報処理装置上にスレッドを分散して実行させる機構をもち、その中で同じ情報処理装置上で動作させるスレッドをスレッド間の関連度に応じて制御するシステムにおいて、分散仮想共有記憶の各ページに対して現在そのページを所有しているスレッドを記憶するステップと、分散タスク内のスレッドの分散仮想共有記憶へのページフォルト情報と、そのページを現在所有しているスレッドの情報とから、フォルトを発声したスレッドとそのページを所有しているスレッドの関連度を決定するステップと、各スレッドに対して他のスレッドの関連度を記録する領域をいくつか用意しそれに記録するステップと、を有することで短時間に同じページにアクセスするスレッドの関連度を高くするような関連度計測が行え、その関連度を記録する記録領域が小さくできる。

【００１３】

【発明の実施の形態】以下、図面を参照して本発明の実施の形態を詳細に説明する。

＜情報処理システムの構成例＞図１は本実施の形態の分散した情報処理装置から成る情報処理システムを示す図である。

【００１４】各情報処理装置１０１は、単独で一般的な情報処理装置として動作することが可能であり、ネットワーク１０２によって接続され相互に通信可能である。但し、各情報処理装置が全ての入出力装置を備えている必要はない。尚、以下の実施の形態では、利用者レベルスレッドを利用者レベルの制御機構によってカーネルスレッドへ割り当てる制御を行っているが、利用者レベルスレッドを持ちいない場合は、カーネルレベルのスレッド移動を行なうスレッド制御機構によって、本実施の形態と同様な効果を得ることができる。

【００１５】図２に本実施の形態のスレッド関連度を用いるスレッド制御機構の概念図を示す。２０１が、各情報処理装置（以下ノードと呼ぶ）で、相互にネットワークによって接続されている。２０２が、オペレーティングシステムのカーネルであり、タスクの制御，そのノード内の主記憶の制御，カーネルレベルのスレッド制御などを行なう。２０３が、複数のノードにまたがる分散タスクである。２０４が、分散タスク内で動作するカーネルスレッドである。２０５が、本スレッド制御方法で利用者レベルスレッドの制御を行なうスレッド管理ライブラリであり、利用者プロセッサにリンクされて利用者レベルで動作する。このスレッド管理ライブラリ２０５によって、カーネルスレッド上で実行する利用者レベルスレッドの選択が行われる。さらに、利用者レベルスレッドの生成，中断，再開などの制御や、ノード間の移動などが行われる。２０６が、ユーザプログラムによって生成された利用者レベルスレッドである。２０７が、分散タスクの分散仮想共有記憶を実現する分散仮想共有記憶管理サーバであり、ここで各スレッドのアクセス情報が収集される。

【００１６】尚、本実施の形態においては、以下に説明するスレッド制御手順はスレッド管理ライブラリ２０５に、スレッド関連度情報及びその制御手順は分散仮想共有記憶管理サーバ２０７にそれぞれ実装されている。これらスレッド管理ライブラリ２０５や分散仮想共有記憶管理サーバ２０７は、各ノードに分散されて配置されていても、特定のノードに集中して配置されていてもよい。また、固定的に配置されても、ネットワーク１０２を介して相互に移動あるいは転送（複写）されてもよい。更に、これら制御手順やデータは、各ノードでオペレーティングシステム等と共にフロッピーディスク等の記憶媒体からロードされるように構成してもよいし、ネットワーク１０２を介してダウンロードされるように構成されてもよい。

＜情報処理システムの動作例＞

（実施の形態１）図３に、本実施の形態において用いられるスレッド制御手順の流れ図を示す。

【００１７】尚、本流れ図の利用者レベルスレッド２０６の切り換えは、プログラム中で自スレッドを明示的にブロックするライブラリ関数を呼び出した場合や、排他制御用のライブラリ関数を呼んだ時にそのロック変数が既にロックされていた場合や、ある条件が成り立つまで待機する条件同期を行なう場合などに、利用者レベルスレッド２０６が自分自身をブロックすることで開始される。以下、流れ図に従って説明する。

【００１８】利用者レベルスレッド２０６の切り換えを

行なおうとする時、まず分散仮想共有記憶サーバ２０４に対して利用者レベルスレッドの関連度情報を要求し、メッセージ転送もしくは共有メモリを用いる等によってスレッド管理ライブラリ２０５内に取り入れる（Ｓ３０）。この利用者レベルスレッド間の関連度情報については後で説明する。さらに、スレッド管理ライブラリ２０５内に存在する動作可能な利用者レベルスレッドのキュー（図示せず）にスレッドが挿入されているかどうか調べる（Ｓ３１）。もし動作可能な利用者レベルスレッドのキューが空で、現在動作可能な利用者レベルスレッドがないならば、現在実行中の利用者レベルスレッドのコンテキスト情報をその利用者レベルスレッドのコンテキスト格納域に保存する。このコンテキスト格納域は、利用者レベルスレッドのデータ構造内もしくは利用者レベルスレッドのスタック等に置かれる。コンテキスト情報を保存した後は、動作可能な利用者レベルスレッドが現れるまで待機状態に入る（Ｓ３２）。

【００１９】逆に動作可能な利用者レベルスレッドのキューにスレッドが挿入されており、現在動作可能な利用者レベルスレッドがあるならば、それらを１個づつ取り出して（Ｓ３３）、その利用者レベルスレッドと現在そのノード（情報処理装置）上で動作している全利用者レベルスレッドとの関連度の総和を計算する（Ｓ３５）。その関連度の総和を動作可能な利用者レベルスレッドのキューに入っている全ての利用者レベルスレッドに関して計算する。

【００２０】キューに入っている全ての利用者レベルスレッドに対して計算がおわったら（Ｓ３４）、そのうちで最も関連度の総和が大きい利用者レベルスレッドを選びだし、動作可能な利用者レベルスレッドのキューから外す（Ｓ３６）。そして、現在動作している利用者レベルスレッドのコンテキストを保存し、選びだした利用者レベルスレッドのコンテキストを読み出して、上記ノードのプロセッサ内のレジスタ等にロードすることで（Ｓ３７）、その利用者レベルスレッドを実行する。

【００２１】図４は、分散仮想共有記憶サーバ２０７内で、各利用者レベルスレッド間の関連度情報を制御するの流れ図である。利用者レベルスレッドが分散仮想共有記憶にアクセスを行なった結果ページフォルトが発生し、分散仮想共有記憶サーバ２０７に通知される。このページフォルトのアドレス等の情報から、分散仮想共有記憶の詳細な状態を保持しているページ管理用のページ管理構造体を得る（Ｓ４０）。ページ管理構造体は、例えば図７でページ所有者(owner threads) が１つの場合の構造を有している。ページ管理構造体には、そのページの分散仮想共有記憶の詳細が記憶されており、それに基づいて、分散仮想共有記憶の一貫性が保証される（Ｓ４１）。尚、分散仮想共有記憶の一貫性が保証については様々な既知の方法があり、ここでは詳細には説明しない。

【００２２】本実施の形態ではさらに、図３のステップＳ３５において利用者レベルスレッドの関連度情報を得るために、以下の操作を行なう。まず、そのページの使用は今回が最初であるか否かを調べる（Ｓ４２）。もし、そのページを最初に使用するのであれば、関連度は変化させずに、ページ管理構造体内にフォールトを発生した利用者レベルスレッド（以下フォールトスレッドとする）を記録して（Ｓ４５）、これ以降の関連度情報の収集に用いる。もし、そのページが既に他の利用者レベルスレッドにて使用中の場合には、ページ管理構造体に現在そのページを所有している利用者レベルスレッド（以下所有スレッドと呼ぶ）が記録されている。その所有スレッドの情報を得て（Ｓ４３）、フォールトスレッドと所有スレッドとの関連度を増加させる（Ｓ４４）。

【００２３】利用者レベルスレッド間の関連度情報５０は、分散仮想共有記憶サーバ２０５内に図５に示されるような配列として存在している。本実施の形態では、各利用者レベルスレッドに対して、関連度情報を保持する利用者レベルスレッドの個数を定めておく。そして、その個数分の関連度情報を保持できるだけの配列を用意する。例えば、図５では、各利用者レベルスレッド５１に対して、関連度が高い方から４つの利用者レベルスレッド５２とその関連度５３とが保持されている。

【００２４】そこで、分散仮想共有記憶のページに対してフォールトが発生した場合に、フォールトスレッドと所有スレッドとの両方のスレッドに対して、図６の流れ図に示される関連度の更新を行なう。まず、フォールトスレッドをスレッドＡ、所有スレッドをスレッドＢとして、図６の流れ図の操作を行い、続いて逆にフォールトスレッドをスレッドＢ、所有スレッドをスレッドＡとして、図６の流れ図の操作を行なう。以下、図６の流れ図に従って説明する。

【００２５】図５に示した関連度情報５０内のスレッドＡに対する関連情報を取り出す（Ｓ６０）。その中に、スレッドＢが含まれているかどうかを調べる（Ｓ６１）。スレッドＢが含まれる場合には、その関連度にａ（予め決定しておいた値）を加える（Ｓ６２）。取り出した関連度情報にスレッドＢが含まれていなければ、関連度情報の配列に空きがあるかどうかを調べ（Ｓ６３）、空きがあればそこにスレッドＢを関連度をａにして記録する（Ｓ６４）。図５の例では、関連スレッドが３つまでなら空きがあることになる。

【００２６】空きがなければ、記録されているうちで関連度が最小のものを捜す（Ｓ６５）。図５のように関連度の大きい順に関連度情報の配列に記憶しておけば、配列の最後に記録されている物が関連度が最小のスレッドとなる。その最小の関連度がａ以下であれば（Ｓ６６）、そのスレッドを削除してスレッドＢと入れ換える（Ｓ６７）。スレッドＢの関連度はａとする。このとき関連度の大きい順にするためには、スレッドＢの挿入位

置を適当に調節して、関連度の順を崩さないようにすれば良い。

【００２７】また、スレッドＡに対して記録されている関連度が全てａよりも大きい場合には、エージングのためにスレッドＡに対する全ての関連度からｂ（あらかじめ決定しておいた値で、ｂ＜ａ）を減ずる（Ｓ６８）。この処理は、この場で行なう方法以外に、全ての関連度情報に対して定期的に定数を減ずる方法等もあり、上記ａ，ｂの値と共にシステムの構成や処理内容等とも関連して、適切なものが選ばれればよい。

【００２８】（実施の形態２）上記実施の形態１に於いては、分散仮想共有記憶の各ページ管理構造体に所有スレッドとして記録されている利用者レベルスレッドと、そのページに対してフォールトを発生したフォールトスレッド間との関連度を増加させて、関連度の制御を行っていた。本実施の形態では、各ページ管理構造体に現在の所有スレッドだけでなく、過去の所有スレッドをも記憶しておき、それらとフォールトスレッド間の関連度も増加させる。

【００２９】図７に本実施の形態で使用されるページ管理構造体７０の例を示す。またこのページ管理構造体７０に所有スレッドを記録する操作の流れ図を図８に示す。図７のように、ページ管理構造体７０に所有スレッドを記録するためのＦＩＦＯ領域７１～７４を用意する。ＦＩＦＯの領域は、所有スレッドが増えた時に動的に確保することも出来る。

【００３０】ＦＩＦＯ領域がある場合は（Ｓ８０，Ｓ８１）、ＦＩＦＯ領域に所有スレッドを記録していき（Ｓ８２，Ｓ８３）、ＦＩＦＯ領域を最初に確保する場合には空きがなくなった場合、動的にＦＩＦＯ領域を確保する場合にはＦＩＦＯがある閾値以上の長さになった場合には、ＦＩＦＯの先頭から最も古くそのページを所有していた利用者レベルスレッドを削除して（Ｓ８４）、最後尾に新たな利用者レベルスレッドを登録する（Ｓ８５）。このような操作を行なって、ページ管理構造体に過去の所有スレッドも記録する。

【００３１】この場合には、時間的に前の所有スレッドと最近の所有スレッドとの差別化のために、関連度を増加させる操作（図６参照）に於いて、実施の形態１で述べたａに時間の新旧に対応してある係数を掛けて重み付けを行ない、その利用者レベルスレッドがどれくらい前にそのページの所有者であったかを反映することも可能である。例えば、所有スレッドがＦＩＦＯに後に挿入された順で線形に重み付けを行うためには、所有スレッドが先頭からＸ番目であったときに、関連度を増加させる値をａ×Ｘ／Ｌ（Ｌは所有スレッドを記録するＦＩＦＯの長さ）とすることで重み付けを行う方法が考えられる。

【００３２】尚、本実施の形態では、利用者レベルスレッドにおける関連度について説明したが、本発明のスレッド関連度の制御はカーネルスレッドに対しても適応ができる。又、本発明は、複数の機器から構成されるシステムに適用しても、１つの機器から成る装置に適用しても良い。また、本発明はシステム或は装置にプログラムを供給することによって達成される場合にも適用できることはいうまでもない。この場合、本発明に係るプログラムを格納した上記の記憶媒体が本発明を構成することになる。そして、該記憶媒体からそのプログラムをシステム或は装置に読み出すことによって、そのシステム或は装置が、予め定められた仕方で動作する。

【００３３】

【発明の効果】本発明により、短時間に同じページにアクセスするスレッドの関連度を高くするような関連度計測が行え、その関連度を記録する記録領域が小さくできるスレッド関連度計測方法を提供できる。又、上記スレッド関連度計測方法を適用してスレッドを制御するスレッド制御方法及び情報処理システムを提供できる。

【００３４】すなわち、分散した複数の情報処理装置をネットワークで接続した分散情報処理装置上に存在するタスクの主記憶を分散仮想共有記憶によって共有する分散タスク内で、各情報処理装置上にスレッドを分散して実行させる機構をもつオペレーティングシステム上の、利用者レベルのスレッドを分散タスク内のスレッドに実行させるスレッド制御機構において、分散タスク内のスレッドの分散仮想共有記憶へのアクセス情報を収集し、収集したアクセス情報からその分散タスク内のスレッド間の相互の関連度を計測し、その関連度を用いて分散タスク内のスレッドを制御することで、分散仮想共有記憶のページスラッシングを低減し、処理能力の低下を防止するようなシステムにおいて、スレッド間の関連度を各スレッド間の２次元の配列に保持しておくのではなく、各スレッドに対して記憶領域を用意し、そこに関連度情報を保持することで、スレッド数が大きくなった場合に関連度情報が膨大になることを防止して記憶領域を効率的に使用することが可能となる。

【００３５】さらに、分散仮想共有記憶の各ページに対して現在そのページを所有しているスレッドを記録しておくだけでなく、過去にそのページを所有していたスレッドを記録しておき、それらのスレッドとそのページに対してフォールトを起したスレッドとの関連度を増加させることによって、より正確にスレッド間の関連度を計測することが可能となる。

【図面の簡単な説明】

【図１】本実施の形態の負荷分散方式を用いる分散した情報処理システムの構成図である。

【図２】本実施の形態のスレッド制御方法におけるプロセスの関係を示す概念図である。

【図３】本実施の形態のスレッド制御方法の手順を示す流れ図である。

【図４】本実施の形態のアクセス情報の収集方法の手順

を示す流れ図である。

【図５】本実施の形態のスレッド間の関連度情報を記憶する配列を示す図である。

【図６】本実施の形態のフォールトが発生した場合に関連度の操作を行う手順を示す流れ図である。

【図７】実施の形態２で使用されるページ管理構造体の例を示す図である。

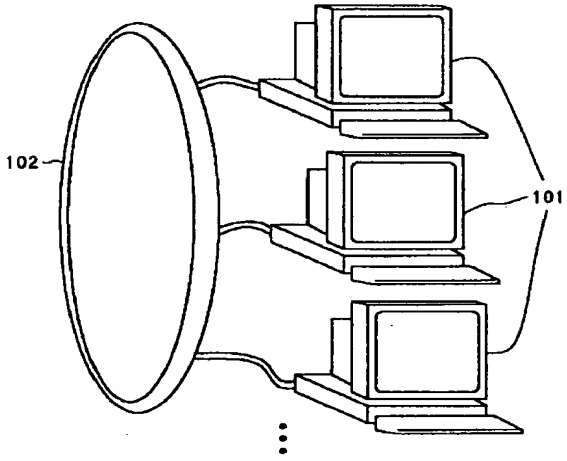
【図８】実施の形態２のスレッドの登録方法の手順を示す流れ図である。

【符号の説明】

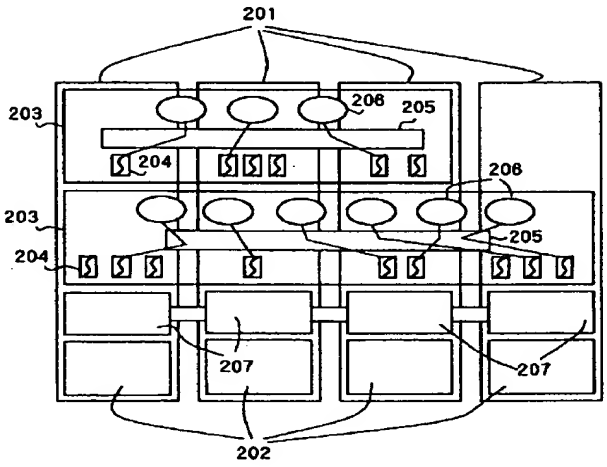
＊１０１　情報処理装置
１０２　ネットワーク
２０１　情報処理装置
２０２　ＯＳカーネル
２０３　分散タスク
２０４　カーネルスレッド
２０５　スレッド管理ライブラリ
２０６　利用者レベルスレッド
２０７　分散仮想共有記憶サーバ

＊10

【図１】



【図２】



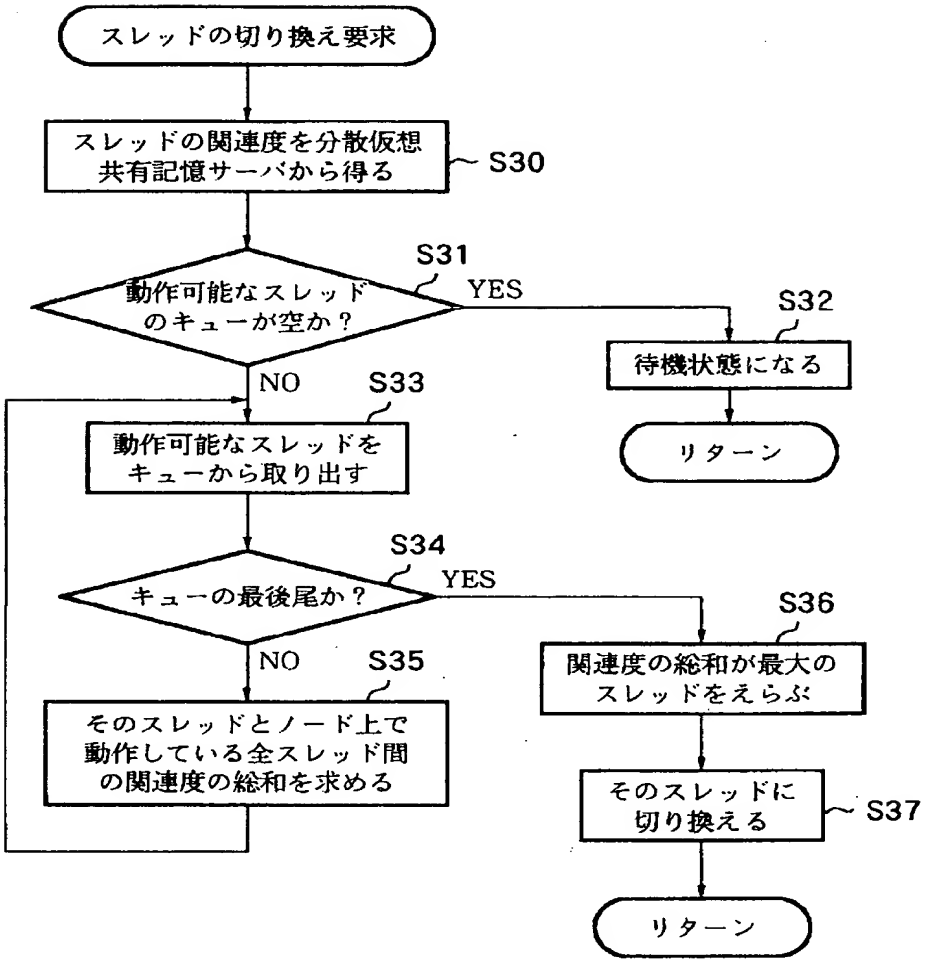
【図５】

大　関連度　小

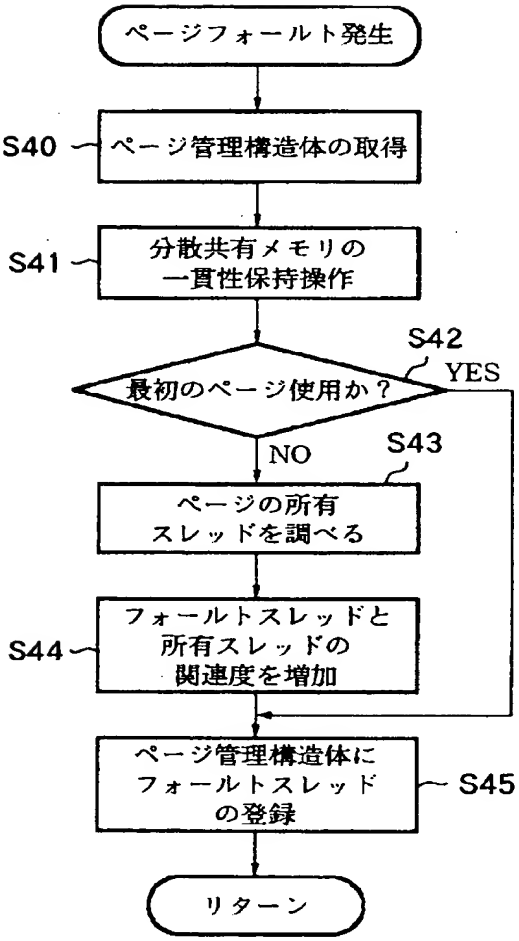
50
51
52
53

スレッド番号	スレッド番号	関連度	スレッド番号	関連度	スレッド番号	関連度	スレッド番号	関連度
1	5	12	4	8	8	4	8	1
2	3	5	6	2	13	1		
3	2	4						
4	1	5	10	4	2	2	8	1
5	1	10	7	5				
6	7	8	5	3	1	1		
7	6	9	12	4				
8	15	14	14	10	9	1		
9	13	12	10	1				
⋮								

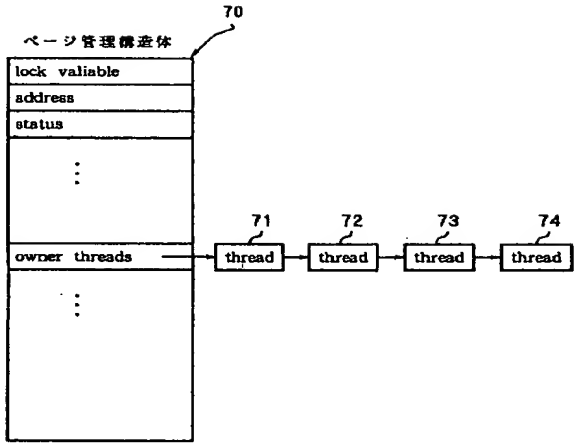
【図３】


スレッドの切り換え要求
スレッドの関連度を分散仮想共有記憶サーバから得る
S30
S31
動作可能なスレッドのキューが空か？
YES
S32
待機状態になる
リターン
NO
S33
動作可能なスレッドをキューから取り出す
S34
キューの最後尾か？
YES
S36
関連度の総和が最大のスレッドをえらぶ
NO
S35
そのスレッドとノード上で動作している全スレッド間の関連度の総和を求める
そのスレッドに切り換える
S37
リターン

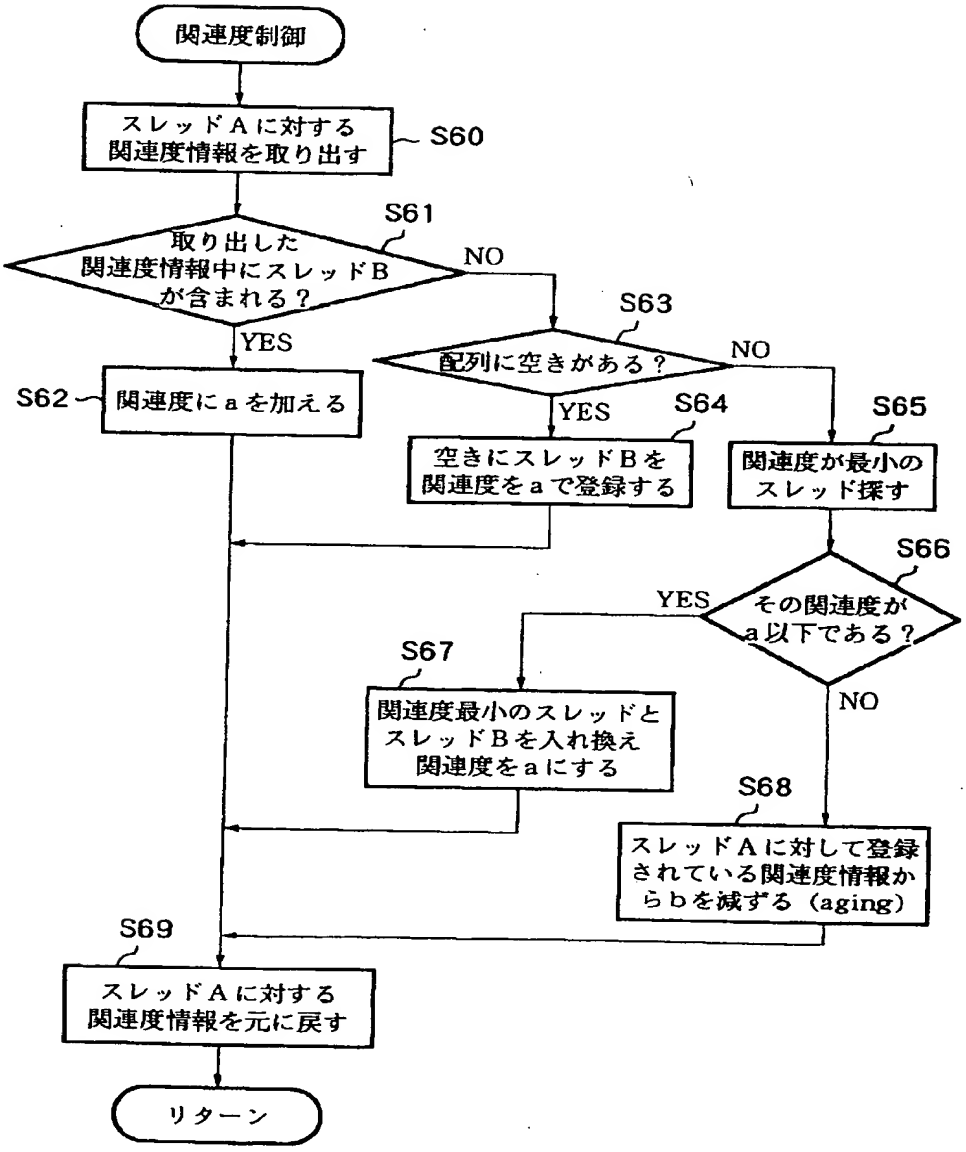
【図４】


ページフォールト発生
S40
ページ管理構造体の取得
S41
分散共有メモリの
一貫性保持操作
S42
YES
最初のページ使用か？
NO
S43
ページの所有
スレッドを調べる
S44
フォールトスレッドと
所有スレッドの
関連度を増加
ページ管理構造体に
フォールトスレッド
の登録
S45
リターン

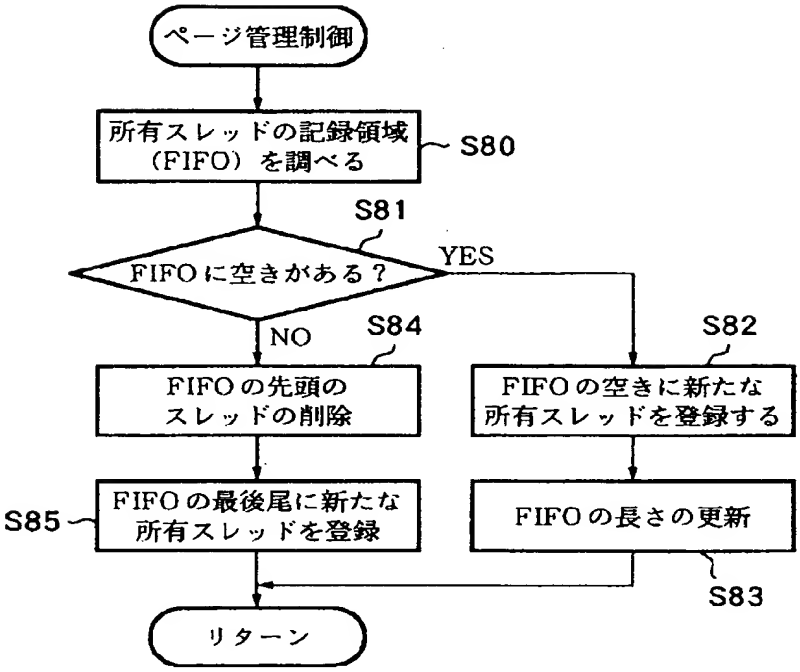
【図７】


70
ページ管理構造体
lock valiable
address
status
owner threads
71
72
73
74
thread
thread
thread
thread

【図６】


関連度制御
スレッドＡに対する関連度情報を取り出す
S60
S61
取り出した関連度情報中にスレッドＢが含まれる？
NO
YES
S62
関連度にａを加える
S63
配列に空きがある？
NO
YES
S64
空きにスレッドＢを関連度をａで登録する
S65
関連度が最小のスレッド探す
S66
その関連度がａ以下である？
YES
NO
S67
関連度最小のスレッドとスレッドＢを入れ換え関連度をａにする
S68
スレッドＡに対して登録されている関連度情報からｂを減ずる（aging）
S69
スレッドＡに対する関連度情報を元に戻す
リターン

【図8】


ページ管理制御
所有スレッドの記録領域（FIFO）を調べる
S80
S81
FIFOに空きがある？
YES
NO
S84
S82
FIFOの先頭のスレッドの削除
FIFOの空きに新たな所有スレッドを登録する
S85
FIFOの最後尾に新たな所有スレッドを登録
FIFOの長さの更新
S83
リターン